

O

AR-009-910

DSTO-TN-0061

T

An Implementation of a Digital
Map Overlay on a Tactical
Display

R. B. Dodd

S

19970307 019

DTIC QUALITY INSPECTED 2

APPROVED FOR PUBLIC RELEASE

© Commonwealth of Australia

D

Mar-05

THE UNITED STATES NATIONAL
TECHNICAL INFORMATION SERVICE
IS AUTHORIZED TO
REPRODUCE AND SELL THIS REPORT

An Implementation of a Digital Map Overlay on a Tactical Display

R.B. Dodd

**Air Operations Division
Aeronautical and Maritime Research Laboratory**

DSTO-TN-0061

ABSTRACT

This report describes a digital map implementation developed for overlaying a tactical information display. The digital map implementation operates over widely varying display ranges, with the ability to render both coastal outlines and shaded landforms. The report discusses features of the implementation approach, data preparation and resultant performance.

RELEASE LIMITATION

Approved for public release

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

Published by

*DSTO Aeronautical and Maritime Research Laboratory
PO Box 4331
Melbourne Victoria 3001*

Telephone: (03) 9626 7000

Fax: (03) 9626 7999

© Commonwealth of Australia 1996

AR No. AR-009-910

October 1996

APPROVED FOR PUBLIC RELEASE

An Implementation of a Digital Map Overlay on a Tactical Display

Executive Summary

Electronic displays are becoming more prevalent in defence platforms and advancements in computer capabilities have made feasible the generation and display of digital map images. This report describes the implementation of a digital map overlay in a simulation and evaluation facility hosted on commercial workstation hardware.

Requirements placed on the implementation included the need to use widely varied display ranges and the ability to display either coastal outlines or shaded landforms. Rendering of digital map images was required to be performed within an allotted time, while resource constraints limited the implementation approaches that could be considered.

A tiled database was used to partition the large amount of coastline data used, permitting rapid access to data for a particular region. Caching of tiled data was used to reduce the incidence of file system accesses. The requirement to use widely varied display ranges made it infeasible to view the same coastline vector data on all ranges since data point densities would be excessive on large display ranges. By clipping and filtering redundant data points prior to rendering, a single coastline database can be used for all display ranges. To render shaded landforms without requiring a separate software module, the implementation uses algorithms which allow identical handling of coastlines and shaded landforms up to the time of rendering.

Performance of the digital map implementation meets all requirements for coastal outline rendering but large range displays of shaded landforms exceed the target rendering time. Filled polygon rendering speeds for the host workstations were measured to be much lower than estimated, causing rendering times to be longer than predicted.

Contents

1. INTRODUCTION	1
2. CONSTRAINING FACTORS	1
3. IMPLEMENTATION MODEL FORMULATION	2
3.1 Tiling scheme	3
3.2 Caching scheme.....	5
3.3 Point filtering algorithm	6
3.4 Clipping algorithm.....	7
4. DATA PREPARATION.....	9
4.1 Data extraction.....	10
4.2 Data transformation	11
5. PERFORMANCE EVALUATION	13
6. CONCLUSIONS	16
7. REFERENCES	16

1. Introduction

Modern defence platforms are being equipped with increasingly advanced sensors and systems that present their crews with large amounts of information on electronic displays. Heavy crew workloads demand that careful consideration be given to crew station designs to ensure that mission critical information can be rapidly located and comprehended. The potential benefits of presenting any additional information need to be weighed against the technical difficulty of generating the information and the impact on crew member performance.

Air Operations Division of the Aeronautical and Maritime Research Laboratory undertakes research into the effectiveness of aircraft crew station designs using facilities developed for the Air Operations Simulation Centre (AOSC). These facilities allow human-in-the-loop simulations to be performed using reconfigurable electronic displays. The Systems Integration Laboratory (SIL), which is part of the AOSC, allows simulation of tactical crew stations and facilitates assessment of tactical system operator station designs under representative workload conditions.

This report discusses the implementation of a digital coastline map on a tactical system operator station display. Tactical system operators utilise graphical displays of sensor detections and contact locations, on which the capability to display coastline maps has long been recognised as desirable. The readily comprehensible nature of a coastline or shaded landform overlay offers improved situational awareness to tactical system operators. Data storage and processing speed requirements for digital maps are very demanding, but inexpensive and powerful computers with large primary and secondary storage capacities make an economical implementation viable. Factors influencing the design, performance and limitations of the implementation are the primary focus of this report, but preparation of coastline data is also discussed.

2. Constraining Factors

A number of constraints were placed on the digital map implementation by the configuration of the SIL. It was necessary to use the XGL¹ high level graphics library already in use for generating other graphical output on the SIL operator stations. This library provides the primitive functions needed to render coastal outlines and shaded landforms, but the display updating mechanism used is not ideal. Double buffering is used to render all display components into a hidden frame buffer before swapping with the visible frame buffer, requiring all graphics to be rendered at every display

¹ XGL is a registered trademark of Sun Microsystems, Inc.

update. Since the rendered map graphic can not be preserved when it does not need updating, rendering time for the map graphic has had to be limited.

The SIL is hosted on SPARCstation² computers containing graphics acceleration hardware capable of rendering 480 000 two dimensional vectors per second. It was desired that display refreshing be possible at a 10 Hz rate, and estimates of processor utilisation resulted in map rendering times of 25 ms being considered an upper limit. Worst case scenarios requiring longer rendering times would be permitted, with the consequence that display update rates may sometimes be less than 10 Hz. A 25 ms rendering time would be sufficient to display 12000 coastline vectors in a single map graphic, many more than are required to generate a detailed coastal outline. Rendering speed for filled polygons was estimated to be five or more times slower than for coastline vectors, so a nominal data point limit was judged to be 2400 points if rendering times for shaded landforms were not to exceed 25 ms.

The display ranges on which the digital coastline map could be viewed span a 500:1 ratio. Displaying high levels of detail over such widely varying display ranges compounds the difficulty of achieving map rendering within the available time. In order to avoid an overly complex implementation, it was considered acceptable to use a coarsely resolved digital coastline on small display ranges where the presentation of minute coastline detail is not of operational benefit. The smallest display range produces a two nautical mile span for a full screen image. It was judged that an acceptable low resolution coastline viewed on this range should have a data point spacing of less than one nautical mile.

Ease of maintenance of the software implementing the digital coastline map was also a consideration. It was desirable that the requirement to render both coastal outlines and shaded landforms not result in the production of two software modules that would each require maintenance as the SIL underwent further development. The preferred approach was to implement a single module that exploited the underlying similarities between the differently rendered graphics.

3. Implementation Model Formulation

The resources required to implement the digital map software and to prepare the coastline data were not to be excessive so a readily implementable approach was required. Processing time required to prepare an image for rendering was not critical since new map images are required relatively infrequently. Rendering times are critical, but can be controlled by limiting the number of data points being used. Use of a single coastline database is preferred over the use of multiple databases of differing

² SPARCstation is a registered trademark of SPARC International Inc.

resolutions, in order to minimise data storage requirements and to reduce the effort required to develop and maintain the database. Initialisation requirements for the digital map are to be minimal so as not to impede rapid reconfiguration of the SIL.

Using the XGL graphics library [1], coastal outlines are rendered as multiple connected line segments and shaded landforms are rendered as filled polygons. Both line segments and polygons utilise lists of points to represent their vertices so that identical processing of both forms of data by a single software module was feasible. Graphics rendering is achieved by high level function calls which accept references to suitably structured data. By storing coastline and shaded landform data in structures directly compatible with XGL library functions, the need to copy data at rendering time is avoided.

The largest display range on which the digital map can be viewed produces a 1024 nautical mile span for a full screen image. Displaying such large areas could require the rendering of tens of thousands of data points if raw map data were used, resulting in excessive rendering times. Filtering and clipping schemes were devised to limit the number of data points being rendered by eliminating redundant points and by clipping points that would not be visible on the display.

The relatively large amount of processing time required to convert coastlines into polygons suitable for rendering shaded landforms led to the adoption of separate databases for coastlines and landform polygons, shifting the processing overhead to the data preparation stage. A tiling scheme was devised for dividing coastline and landform data into manageable partitions and a caching scheme was used to reduce the number of file system read operations performed as the digital map spanned different regions during a simulation run.

3.1 Tiling scheme

The size of a world wide database precludes the use of an all encompassing coastline data file and leads to the use of a tiled database. It is desirable that the digital map overlay operate over any region without the need for priming or region specific initialisation, and this is readily accommodated using a tiled database. Loading of coastline data for any displayed region only requires that data be loaded from all tiles that are exposed in the region. Selection of the tile size to use is not fixed by the design of the implementation, allowing measured run time performance to be used to select an optimum tile size. The indexing scheme used for locating tiled data files uses tile positions relative to the intersection of the equator and the prime meridian in a tile size independent manner.

Tiles loaded to span a displayed region normally contain data exceeding the extent of the displayed region, and this data requires clipping to preserve rendering efficiency.

The larger the tile size employed, the more data exceeding the extent of the displayed region may be loaded, and the more clipping is required. The left hand image in Figure 1 shows four data tiles loaded to span a displayed region indicated by a dashed line, with coastline data requiring clipping being shown as a dotted line. The right hand image in Figure 1 shows the nine smaller data tiles required to span the displayed region and indicates with dotted lines the data tiles not loaded compared to when larger tiles are used. As a consequence of less unnecessary data being loaded, the right hand image in Figure 1 can also be seen to contain less coastline data requiring clipping.

Although small tile sizes reduce clipping requirements, other effects of tile size need to be considered. Tile sizes which are small compared to the displayed region require the loading of many small data files which increases the time required to load data spanning a region. Tile size also determines the maximum length of coastline segment likely to be encountered in the loaded data, with small tiles resulting in the loading of many short coastline segments rather than fewer, longer coastline segments. Rendering speeds are higher for longer coastline segments due to the reduced number of individual line segments requiring processing. When shaded landforms are being rendered the tile size determines the number of inland polygons used to shade inland areas. Small tiles require the use of many small inland polygons while large tiles require fewer large inland polygons which are rendered more efficiently.

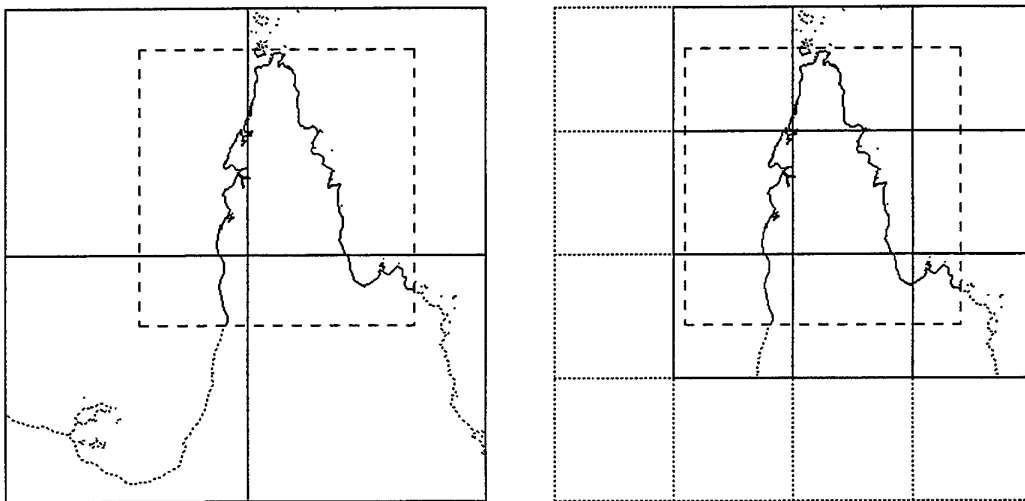


Figure 1. Clipping requirements for two choices of tile size.

3.2 Caching scheme

For a given display region, data is initially loaded for all map tiles that are exposed. As the displayed region changes due to platform movement, some tiles become hidden and others are exposed. Since loading data from a file is much slower than accessing data held in primary storage, it is desirable to minimise the number of file loads performed. In missions where platform course is frequently changing, it is possible that loaded tiles which have become hidden will be exposed again, as is certainly the case when flying any form of search pattern. By holding data from previously exposed tiles in a cache, the number of file system accesses can be reduced.

The cache is modelled as a two dimensional array of primary storage for data files and can be considered to span a rectangular area. Data for a particular tile is only loaded into the cache when the tile becomes at least partly exposed in the displayed region. Since the cache spans a contiguous set of data tiles, loading a new data tile may require that the cache boundary be moved to include the new data tile. As a consequence of cache boundary movement, data tiles furthest from the new data tile may now fall outside the cache boundary and be released from the cache.

Figure 2 shows the changes made to cache contents when loading newly exposed data tiles requires movement of the cache boundary and results in the release of some cached data. The dashed line identifies the displayed region, within which all exposed tiles must be loaded into the cache. Shaded tiles represent cache locations for which data has not been loaded. Starting from the displayed region indicated in the left hand image, the displayed region is moved to the right past the cache boundary to result in the cache contents shown in the right hand image.

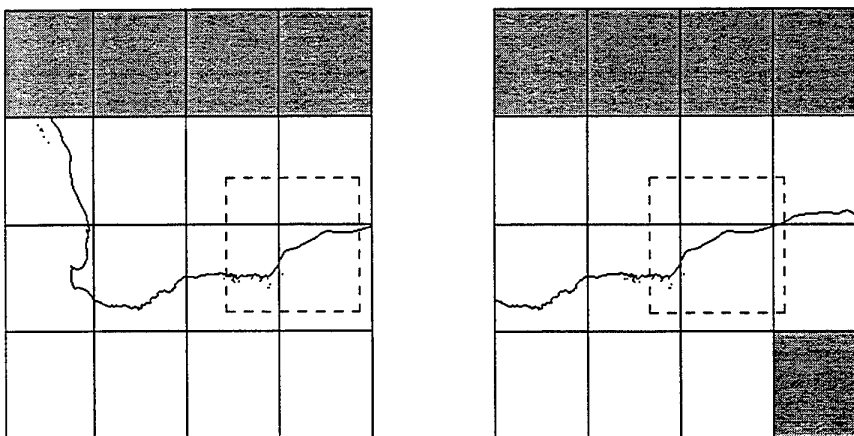


Figure 2. Cache contents before and after a change of viewing area.

3.3 Point filtering algorithm

The 500:1 ratio between the largest and smallest display ranges on which the digital coastline map can be viewed makes it infeasible to view the same map data on all ranges. On large ranges the number of data points rendered would exceed the 2400 point limit intended to limit rendering times to acceptable values, with many points mapping to identical pixels on the display device. A requirement for the implementation is the ability to efficiently filter out redundant data points at run time. This filter needs to be applicable to both coastal outlines and shaded landforms, which requires preservation of closure and coverage properties of landform polygons.

By using coastline data with near constant spacing between data points, a simple but effective form of point filtering is possible. On display ranges for which adjacent data points map to the same display pixel, discarding redundant points reduces the data point count without affecting the shape of the rendered coastline. Only for cases where the actual data point spacing is larger than expected do discrepancies in the shape of the rendered coastline occur. If it was critical that the coastline be resolved to the exactly correct pixels, interpolation between coastline points could have been used to generate constant spaced data during data preparation. On large display ranges this filter limits the number of points displayed in relation to the number of pixels on the display device, although in practice more coastlines are exposed on larger ranges and so data point counts still increase.

Using this filtering scheme when filled polygons are being rendered to create shaded landforms introduces additional considerations. In a tiled landform database, the polygons forming a shaded landform must meet along all inland edges to ensure complete coverage for landforms to be rendered without gaps between component polygons. In the simple case where a coastline enters one side of a tile and exits the other side, a landform polygon is created by closing the coastline around the inland edges of the tile boundary. Filtering of coastline points within the landform polygon still offers the benefit of eliminating points that would map to the same display pixel. However, closure points added to the coastline to form the landform polygon are not constant spaced data and if discarded would destroy the complete inland coverage of the polygon. Hence, four points at each end of a data point list are not regarded as constant spaced data and are not candidates for being discarded. The penalty of this approach is that coastal outlines will contain extra points at the ends of their point lists that could otherwise have been filtered out.

Small offshore islands account for a significant proportion of the coastal points in many regions, and warrant special attention since they present a difficulty for the filtering algorithm described thus far. By assuming that all coastline segments contain four closure points at each end which can not be discarded, small islands will contain relatively large numbers of points that can not be discarded and the filtering algorithm

is ineffective. Detection of coastline segments forming closed islands is easily facilitated by constructing islands to have coastline segments that start and end at identical points. Having a simple test for detecting closed islands allows them to be treated as special cases. No closure points exist in island coastlines, so all points are candidates for being discarded and the filtering algorithm is again effective. As the display range is increased many islands are reduced to single pixels on the display device, regardless of further increases in display range, so that filtering efficiency is again degraded. By allowing islands that map to a small percentage of a display pixel to be discarded completely, filtering efficiency is regained for such cases.

3.4 Clipping algorithm

Loading coastline or shaded landform data for tiles exposed in a displayed region will include data that exceeds the extent of the region. To reduce rendering times an initial clipping process is used to lessen the amount of clipping performed at rendering time to a relatively low level. An algorithm for rapidly clipping landform polygons without affecting polygon coverage in the displayed region, and which still provides satisfactory clipping for coastal outlines, is discussed in the following paragraphs.

Determining whether a point outside the displayed region can be removed without affecting polygon coverage within the region is performed using the infinitely extended boundaries of the region. By clipping such that only coastline edges to the left of the displayed region can be affected, it is not possible for polygon coverage within the region to be altered. By similarly allowing clipping of edges that are entirely above, below or to the right of the displayed region, the majority of coastline edges or landform polygons can be rapidly clipped while polygon coverage inside the region is preserved. Clipping is only performed against one boundary since the majority of candidate points are typically removed in a single clipping pass. Coastline segments considered candidates for clipping are those with one or both ends originating outside of the displayed region and clipping is started from these end points.

Operation of the clipping algorithm can be explained using the example illustrated in figure 3, where the displayed region is indicated by dashed boundary lines. The closed polygon in the left hand image of figure 3 has coincident start and end points labelled as vertex A. To clip this polygon, two clipping operations are initiated at vertex A, along the starting and ending edges of the polygon. Since all clipping must be performed to one side of the displayed region, the presence of polygon closure points which trace the boundary of the enclosing tile would often halt clipping after only a few points are traversed. This problem is avoided by considering the four vertices at each end of a polygon to be closure points which are traversed before clipping is initiated.

The first point encountered after the closure points in figure 3 is above the displayed region, so clipping is performed against the upper boundary of the region. Points are discarded up to, but not including, the last point that is above the upper boundary of the displayed region, labelled as vertex *B*. The newly created edge between the last closure point and a point above the displayed region replaces the discarded edges without affecting polygon coverage within the region. For clipping to the right of the displayed region in figure 3, polygon edges are traversed from vertex *A* past the closure points to vertex *D* where coastline points begin. Coastline points up to, but not including, vertex *C*, the last point to the right of the displayed region are discarded to create a single edge. The right hand image in figure 3 shows the clipped polygon and its unchanged coverage within the displayed region.

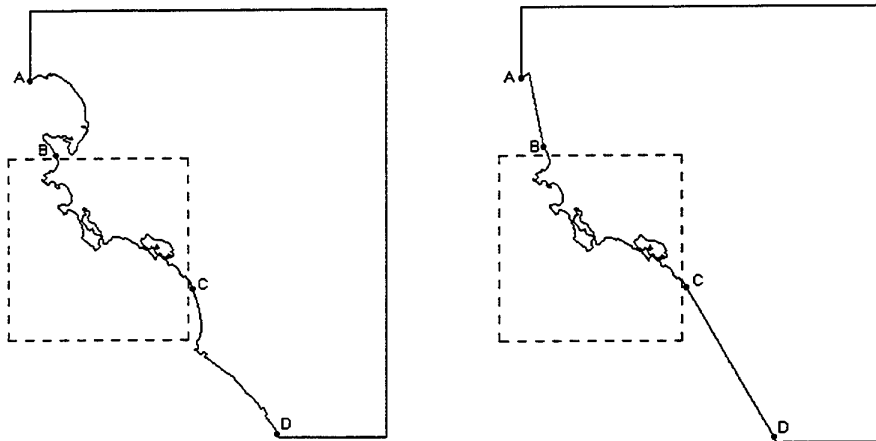


Figure 3. A landform polygon before and after clipping.

To further illustrate how the clipping algorithm preserves the coverage of polygons within the displayed region consider the left hand image in figure 4. The coincident polygon start and end points are labelled as vertex *A*. Preserving closure points and then discarding points up to, but not including, vertex *B*, the last point outside of the displayed region results in the polygon depicted in the centre image in figure 4. Note that the edge replacing discarded edges intersects the displayed region and erroneously creates an inland corner within the region. If the algorithm was followed correctly, points would be discarded up to the point preceding vertex *C*, the last point to the left of the infinitely extended left boundary of the displayed region, and the result shown in the right hand image of figure 4 would be obtained. Some coastline points outside the displayed region are not clipped in this situation, but the resulting polygon does not have its coverage altered within the displayed region.

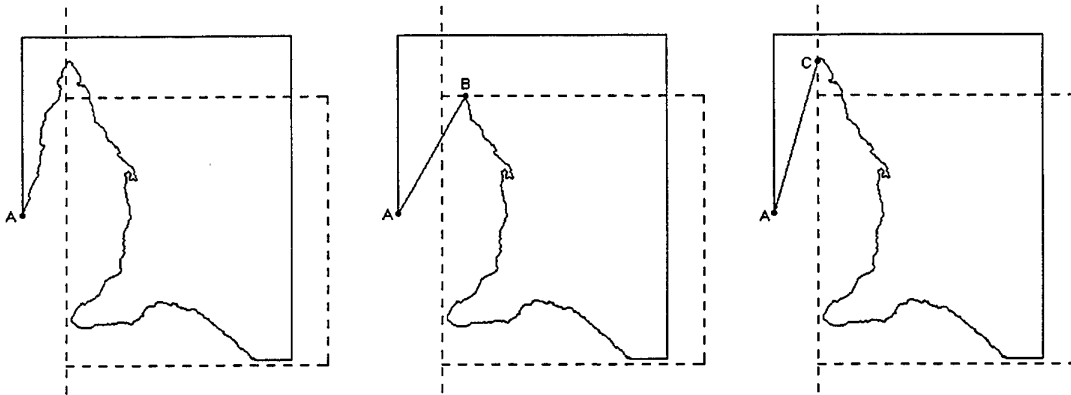


Figure 4. A landform polygon before clipping, incorrectly clipped and correctly clipped.

4. Data Preparation

Preparation of coastline and shaded landform data was a significant component of the digital map implementation. To facilitate the automated generation of landform polygons from coastline data it was necessary to identify the inland side of all coastline segments used as raw data. The source for collection of raw data was the Digital Chart of the World (DCW) geographic database³.

The DCW geographic database is constructed from the cartographic negatives used to create 1:1000000 scale Operational Navigation Charts (ONC), digitised at a resolution of 500 Dots Per Inch (DPI). Post digitisation processing introduces topological relationships into the database. Digitisation of an ONC at 500 DPI provides point spacings of less than one nautical mile and meets the resolution requirements for the digital map data, although optimisations have been performed that eliminate collinear points.

Extraction of feature data from the DCW database is only indirectly supported by software delivered with the database, requiring development of a data extraction mechanism. Database tables are accessed to extract coastline data from the database and further processing is then used to transform the raw DCW data into a suitable format for the digital map. Alternatively, source code supplied for a program to view

³ The DCW database is available to the civilian community from the Australian Surveying and Land Information Group (AUSLIG), a business unit within the Commonwealth Department of Administrative Services. Military availability should be referred in the first instance to the Army Survey Directorate (DSVY-A).

the DCW database could have been adapted to read from the database and generate output in a suitable format.

4.1 Data extraction

The database architecture employed by the DCW geographic database is Vector Product Format (VPF) [2]. The VPF database model is specifically designed for geographic information systems and combines relational data model concepts with planar topological objects to model nodes, edges and faces for features on a geographic coverage. Simple schemas are used to define tables which may store instances of feature attributes and foreign keys to other tables in order to implement relationships between table instances. The DCW database also uses the VPF option of database tiling to limit table sizes.

In the VPF database model, coastline data is stored as edges defining a boundary between areas of land and sea which are represented as faces. For a coverage there exists a line feature table and an area feature table identifying the types of all edges and faces in the coverage. Faces to the left or right of a given edge can be determined to be either land or sea, and the simplest way of encoding this information in the extracted data is to arrange the ordering of coastline edges so that coastlines progress anti-clockwise around the land masses that they define. Utility software provided with the DCW database is able to dump the contents of any table to a text file. This utility provides a means for accessing data tables and was employed to extract coastline data from the database.

After selecting a geographic area from which to collect coastline data, a batch processing file is created which invokes the dump utility to write to data files the contents of all face and edge tables spanned by the area. By similarly writing the contents of the line and area feature tables for the coverage to data files, extraction of coastline data is possible by resolving relationships between records in these tables.

For each edge in the edge table of a database tile, foreign keys to the face table for the tile are used to determine the left and right face foreign keys to the area feature table for the coverage. The edge table foreign keys to the line feature table are then used to identify edge types, and edges which are not coastlines or coastal closure lines are discarded. For the remaining edges the left and right face foreign keys to the area feature table are used to determine the inland side of the edge, and if necessary the order of the points defining the edge is reversed.

4.2 Data transformation

After extracting directed coastline segments for a geographic area, the first processing step applied is to partition the data into tiles of the size selected for the tiling scheme. This tiling scheme is independent of the DCW tiling scheme. Coastline segments crossing a tile boundary are broken at the tile boundary with new points being added at the boundary crossing in each tile to maintain continuity of the coastline up to the edge of each tile.

Coastline segments within each tile are then concatenated into longer coastline segments where possible. This step is desirable because a small number of long coastline segments can be rendered faster than a larger number of short coastline segments. Additionally, an overlapping point present at the end of a coastline segment which is repeated at the beginning of the continuing coastline segment is eliminated, offering significant savings when many short coastline segments are concatenated. The ideal result for this step is the reduction of the data within a tile to a single line segment, but there are many cases where this is not possible. The coastline may enter and leave a tile several times creating disjoint coastline segments, and offshore islands also result in disjoint coastline segments.

Production of polygon data for rendering shaded landforms requires further processing. Landform polygons are constructed by closing coastline segments along inland tile boundaries, subject to several rules. Inland tiles which do not contain any coastline points are assigned a single landform polygon tracing the tile boundary in order to provide shading for inland areas. In tiles containing a single coastline segment, points are added to close the inland side of the coastline along the tile boundary. Such a tile is illustrated in the left hand image of figure 5 with the necessary closure line shown as a dotted line.

When several disjoint coastline segments exist within a tile, additional rules govern polygon construction in order to maintain compatibility with the point filtering algorithm which requires that closure points only be added to the ends of a coastline segment. In tiles containing one open coastline segment accompanied by one or more closed offshore islands, points are added that close the open line segment along the tile boundary. An example of such a tile is presented in the centre image of figure 5. When there are two or more open coastline segments in a tile the nature of these segments determines the method of achieving closure. If all open coastline segments can be closed along the tile boundary without becoming connected to any other coastline segment, then landform polygons are formed as shown in the right hand image of figure 5. If two or more disjoint coastline segments would become connected as closure points are added along the tile boundary, the requirement to only insert closure points at the ends of line segments is not satisfied and an alternative approach is required.

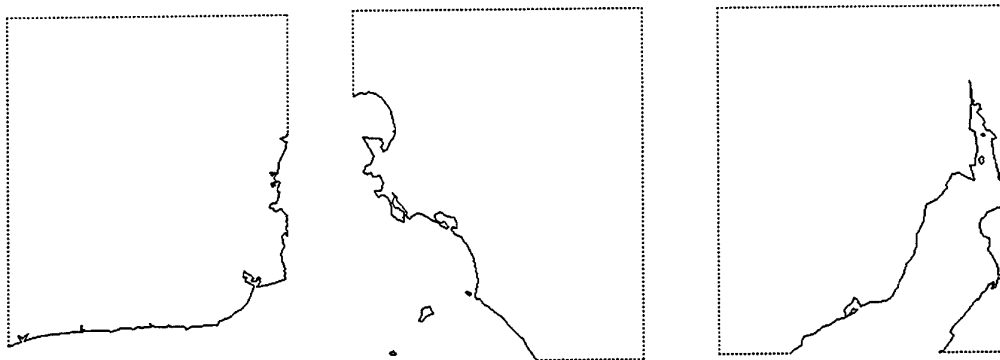


Figure 5. Tiles for which coastline closure is possible along the tile boundary.

For tiles where coastline segments can not be closed along the tile boundary without becoming connected, the tile is subdivided into four quadrants and the closure algorithm is applied to each quadrant. Inland quadrants are assigned single landform polygons tracing the quadrant boundary. The smaller area spanned by a quadrant is likely to contain less disjoint coastline segments so there is an increased likelihood of being able to construct landform polygons for quadrants. Those quadrants for which closure of open coastline segments can still not be achieved are recursively subdivided until landform polygons can be constructed. Recursive subdivision is guaranteed to achieve the desired result since coastline segments do not intersect and by selecting a small enough area no more than one coastline segment will be found, in which case a landform polygon can always be constructed. Examples of tiles needing subdivision to form landform polygons are shown in figure 6. While rendering efficiency is reduced when small polygons are used in place of larger polygons, the adaptive nature of the closure algorithm only subdivides areas that can not be resolved into landform polygons.

Careful examination of the right hand image in figure 6 illustrates an inherent weakness in the subdivision algorithm used to facilitate construction of landform polygons. While subdivision is required to separate the two coastline segments terminating at the upper tile boundary, side effects are introduced within other areas of the tile. Subdivision of the upper right quadrant of the tile has created additional areas containing multiple open coastline segments that can not be closed to form landform polygons without further subdivision. The presence of concave bays or branching inland estuaries, rather than a smooth convex coastline, introduces the risk that a subdivided area may contain several open coastline segments, and hence may require further subdivision for landform polygons to be created. While the algorithm

does not fail to produce correctly constructed landform polygons, it may result in the use of additional subdivision and smaller polygons than ideally necessary.

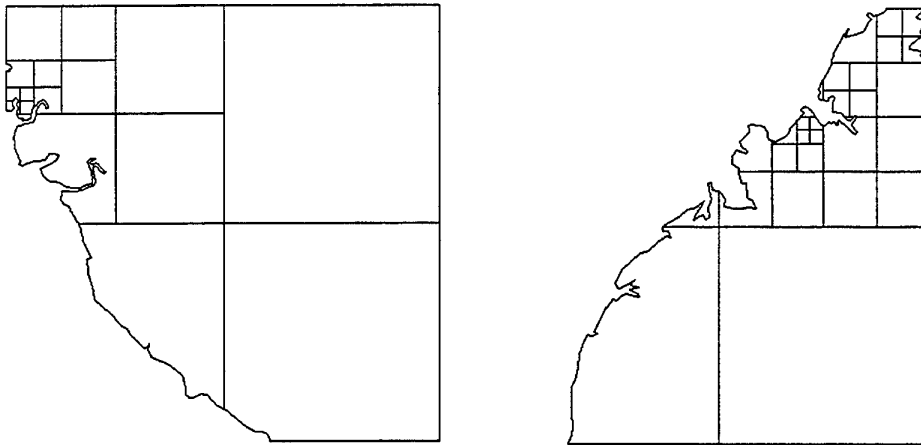


Figure 6. Example tiles requiring subdivision before landform polygons could be formed.

5. Performance Evaluation

SIL run time performance is highly dependent on graphic rendering speeds, allowing rendering speed to be used to assess performance of the digital map overlay. A nominal maximum rendering time of 25 ms for coastal outlines or shaded landforms was the target for the implementation, but has not been satisfied for all conditions. Coastal outlines can be rendered within this time but landform polygons may require significantly more time. Favourable, typical and worst case examples are used to illustrate the rendering times achieved and factors contributing to measured performances are identified.

The actual number of points needed to represent coastlines or shaded landforms for a given display range varies widely depending on the particular area being viewed. The southwest coastline of Australia requires relatively few points to represent due to the smooth coastline and lack of offshore islands, while northern Australia has a more convoluted coastline with numerous offshore islands requiring many more points to represent. Using the maximum 1024 nautical mile display range allows worst case rendering times to be assessed, but typical usage involves smaller display ranges where reduced numbers of points are rendered. On small display ranges of 64 nautical miles or less, rendering performance is generally better than on larger ranges due to the reduced numbers of points involved. Three areas around Australia having

relatively low, medium and high densities of coastal points were used for performance evaluation. Table 1 shows the numbers of coastline and landform polygon points contained in the tiles spanned by the three test areas for large display ranges.

Table 1. Data point counts for three test areas and four display ranges.

Test Area	Coastline Points				Landform Polygon Points			
	1024 n mile	512 n mile	256 n mile	128 n mile	1024 n mile	512 n mile	256 n mile	128 n mile
South Western Australia	7121	3801	1738	1738	7572	4160	2091	2091
South Eastern Australia	13 611	6296	3272	753	16 391	8626	3831	908
Northern Australia	41 367	15 614	7375	4169	50 345	19 921	8134	4171

As indicated by table 1, the number of data points exposed when using large display ranges can exceed the 2400 point target by more than a factor of twenty times. Data point counts for landforms can be seen to be typically twenty percent larger than for coastlines due to the presence of inland tiles and the partitioning of some tiles into smaller regions where multiple inland polygons are required. Since rendering times for coastal outlines are directly proportional to the number of vectors to be drawn, the effectiveness of the clipping and filtering algorithms determined whether the performance target could be achieved. Rendering times for landform polygons are less predictable since the number of data points, the shapes of the component polygons and the proportion of the display surface covered by land all interact in a relationship with rendering time. Effectiveness of the point filtering and clipping algorithms can be assessed using table 2 to compare data point counts for coastlines and landform polygons after clipping and filtering with the initial data point counts in table 1.

Table 2. Post clipping and filtering data point counts for three test areas and four display ranges.

Test Area	Coastline Points				Landform Polygon Points			
	1024 n mile	512 n mile	256 n mile	128 n mile	1024 n mile	512 n mile	256 n mile	128 n mile
South Western Australia	556	414	195	162	1071	883	192	159
South Eastern Australia	1204	915	550	305	4946	2907	753	353
Northern Australia	4545	1624	1572	831	16 004	2927	1904	831

Data point counts in table 2 can be seen to meet the 2400 point target for coastal outlines except for maximum range viewing of worst case data. For landform polygons there are several cases where large ranges or large numbers of tiled database points result in the target point count being exceeded. It can be seen that clipping and filtering of landform polygons is less effective in reducing data point counts than clipping and filtering of coastline data. This difference can be attributed to the presence of inland tiles and subdivided tiles containing multiple inland polygons, which are not candidates for modification by the filtering algorithm.

Rendering times for the three test areas using large display ranges are summarised in table 3. Note that the rendering speed for coastline vectors is in some cases better than the claimed two dimensional vector rendering speed of 480 vectors per ms for the host computer. This is due to the prevalence of very short vectors, many of which may be only two pixels long. Worst case coastline rendering is achieved in less than one third of the allotted time. Rendering times for landform polygons are longer than anticipated, exceeding the nominal 25 ms limit in many cases. Measured rendering speeds for landform polygons are not dependent on data point count alone, but a mean rendering speed of 28 points per ms was calculated for the three test areas to establish a measure of relativity to the rendering speed for coastlines. This mean polygon rendering speed is 19 times slower than the rendering speed for coastlines. When combined with the facts that landform data contains twenty percent more points than coastline data and that clipping and filtering are only half as effective for landform data, the relative speed of landform rendering, which is on average 49 times slower than coastline rendering, is explained.

Table 3. Rendering times for three test areas and four display ranges.

Test Area	Coastline Rendering Times (ms)				Landform Polygon Rendering Times (ms)			
	1024 n mile	512 n mile	256 n mile	128 n mile	1024 n mile	512 n mile	256 n mile	128 n mile
South Western Australia	1.09	0.85	0.52	0.49	43.1	40.7	17.2	14.8
South Eastern Australia	2.19	1.66	1.07	0.72	150	101	31.5	17.3
Northern Australia	8.08	2.90	2.85	1.58	494	115	105	48.5

6. Conclusions

The implementation of the digital coastline map has met the requirement to render both coastal outlines and shaded landforms using a single software module. Rendering times for coastal outlines are better than required but rendering times for shaded landforms in unfavourable scenarios exceed the target rendering time. The initial estimate for the rendering speed of landform polygons was five or more times slower than coastline rendering, but this was much lower than the measured 49 times difference. If a better estimate of the speed difference had been made then a different approach would have been developed to permit shaded landform rendering within the available time.

Algorithms developed to eliminate redundant data points and to enable rapid clipping of edges without affecting the coverage of landform polygons within the displayed region have been demonstrated to perform adequately. Recursive subdivision of tiles to facilitate formation of landform polygons results in the generation of smaller landform polygons than are ideally necessary in some circumstances, but does not fail to produce valid landform polygons and does not perform unnecessary subdivisions if coastline segments can be closed.

Use of relatively low cost hardware and an uncomplicated implementation has demonstrated that digital map generation is feasible without incurring great expense. For an implementation not requiring rendering of the digital map at every display update, or with a smaller variation of possible display ranges, it is likely that the implementation model developed here could be adapted to deliver acceptable run time performance under all conditions.

7. References

1. Sun Microsystems Incorporated, *Solaris XGL 3.0 Programmer's Guide*, 2550 Garcia Avenue, Mountain View, California, 94043-1100, 1992.
2. USA Dept. of Defence, *Military Standard - Vector Product Format*, MIL-STD-600006, Naval Publications and Forms Centre, 5801 Tabor Avenue, Philadelphia, PA 19120-5099, April 1992.

DISTRIBUTION LIST

An Implementation of a Digital Map Overlay on a Tactical Display

R.B. Dodd

AUSTRALIA

1. DEFENCE ORGANISATION

a. Task Sponsor

b. S&T Program

Chief Defence Scientist	} shared copy
FAS Science Policy	
AS Science Industry and External Relations	
AS Science Corporate Management	
Counsellor Defence Science, London (Doc Data Sheet)	
Counsellor Defence Science, Washington (Doc Data Sheet)	
Scientific Adviser to MRDC Thailand (Doc Data Sheet)	
Director General Scientific Advisers and Trials/Scientific Adviser Policy and Command (shared copy)	
Navy Scientific Adviser (3 copies Doc Data Sheet and 1 copy distribution list)	
Scientific Adviser - Army (Doc Data Sheet and distribution list only)	
Air Force Scientific Adviser	
Director Trials	

Aeronautical and Maritime Research Laboratory
Director

Electronics and Surveillance Research Laboratory
Director

Chief of Air Operations Division
Research Leader, Avionics
Head, Avionics Technology
Author, R.B. Dodd

DSTO Library
Library Fishermens Bend
Library Maribyrnong
Library DSTOS (2 copies)
Australian Archives
Library, MOD, Pyrmont (Doc Data Sheet only)

c. Forces Executive

Director General Force Development (Sea) (Doc Data Sheet only)
Director General Force Development (Land) (Doc Data Sheet only)

e. Army

ABCA Office, G-1-34, Russell Offices, Canberra (4 copies)

g. S&I Program

Defence Intelligence Organisation
Library, Defence Signals Directorate (Doc Data Sheet only)

i. B&M Program (libraries)

OIC TRS, Defence Central Library
Officer in Charge, Document Exchange Centre (DEC), 1 copy
*US Defence Technical Information Centre, 2 copies
*UK Defence Research Information Center, 2 copies
*Canada Defence Scientific Information Service, 1 copy
*NZ Defence Information Centre, 1 copy
National Library of Australia, 1 copy

2. UNIVERSITIES AND COLLEGES

Australian Defence Force Academy
Library
Head of Aerospace and Mechanical Engineering
Senior Librarian, Hargrave Library, Monash University
Librarian, Flinders University

3. OTHER ORGANISATIONS

NASA (Canberra)
AGPS

OUTSIDE AUSTRALIA

4. ABSTRACTING AND INFORMATION ORGANISATIONS

INSPEC: Acquisitions Section Institution of Electrical Engineers
Library, Chemical Abstracts Reference Service
Engineering Societies Library, US
American Society for Metals
Documents Librarian, The Center for Research Libraries, US

5. INFORMATION EXCHANGE AGREEMENT PARTNERS

Acquisitions Unit, Science Reference and Information Service, UK

Library - Exchange Desk, National Institute of Standards and Technology, US

ADDITIONAL DISTRIBUTION

CO, Electronic Warfare SQN

HQADF, DD-EW

RANTEWSS, OIC

Director of Air Warfare

AWWSEW in DAW

SPARES (10 copies)

Total number of copies: 58

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION DOCUMENT CONTROL DATA				1. PRIVACY MARKING/CAVEAT (OF DOCUMENT)					
2. TITLE An Implementation of a Digital Map Overlay on a Tactical Display				3. SECURITY CLASSIFICATION (FOR UNCLASSIFIED REPORTS THAT ARE LIMITED RELEASE USE (L) NEXT TO DOCUMENT CLASSIFICATION) Document (U) Title (U) Abstract (U)					
4. AUTHOR(S) R.B. Dodd				5. CORPORATE AUTHOR Aeronautical and Maritime Research Laboratory PO Box 4331 Melbourne Vic 3001					
6a. DSTO NUMBER DSTO-TN-0061		6b. AR NUMBER AR-009-910		6c. TYPE OF REPORT Technical Note		7. DOCUMENT DATE October 1996			
8. FILE NUMBER M1/9/303		9. TASK NUMBER 95/180		10. TASK SPONSOR DGFD (Aerospace)		11. NO. OF PAGES 16		12. NO. OF REFERENCES 2	
13. DOWNGRADING/DELIMITING INSTRUCTIONS None				14. RELEASE AUTHORITY Chief, Air Operations Division					
15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT <i>Approved for public release</i> OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE CENTRE, DIS NETWORK OFFICE, DEPT OF DEFENCE, CAMPBELL PARK OFFICES, CANBERRA ACT 2600									
16. DELIBERATE ANNOUNCEMENT No limitations									
17. CASUAL ANNOUNCEMENT Yes									
18. DEFTEST DESCRIPTORS Computer Modelling Simulation, Systems Integration Laboratory, Air Operations Simulation Centre									
19. ABSTRACT This report describes a digital map implementation developed for overlaying a tactical information display. The digital map implementation operates over widely varying display ranges, with the ability to render both coastal outlines and shaded landforms. The report discusses features of the implementation approach, data preparation and resultant performance.									